UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/695,889	10/28/2003	Gautam Ghose	00121-000700000	7789

7590 09/04/2008

Law Office Of Leland Wiesner
1144 Fife Ave.
Palo Alto, CA 94301

EXAMINER	
CHU, GABRIEL L	

ART UNIT	PAPER NUMBER
2114	

MAIL DATE	DELIVERY MODE
09/04/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/695,889
Filing Date: October 28, 2003
Appellant(s): GHOSE ET AL.

Leland Wiesener
Reg. No. 39424
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 26 June 2008 appealing from the Office action mailed 20 November 2007.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

5666481	LEWIS	9-1997
2002/0019922	REUTER ET AL.	2-2002

6629266	HARPER ET AL.	9-2003
6336139	FERIDUN ET AL.	1-2002
6446218	D'SOUZA	9-2002

"threshold" from the IEEE dictionary

"graphical user interface" from the Microsoft Computer Dictionary

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

1. **Claims 1-4, 6-11, 18-21, 23-26, 33 rejected under 35 U.S.C. 103(a) as being unpatentable over US 5666481 to Lewis in view of US 20020019922 to Reuter et al. and US 6629266 to Harper et al.**

2. Referring to claim 1, 18, 33, Lewis discloses initializing a primary failure analysis module for processing error events and error actions (Wherein a functioning process must have been initialized at some point.);

identifying one or more predetermined error actions and one or more error events associated with the network (Figure 4, elements 100, 102, 104. From the abstract, “network”.);

specifying, according to one or more rules, an error pattern based upon a combination of one or more error events in the storage area network (Figure 4, elements 100, 102);

and associating an error action to perform, according to the one or more rules, in response to receiving the combination of one or more error events of the error pattern (Figure 4, elements 102, 104, 106).

Although Lewis does not specifically disclose that the network may be a storage area network and that the errors are managed by a storage virtualization controller, error handling by a storage virtualization controller in a SAN is known in the art. An example of this is shown by Reuter, see paragraphs 23-26, wherein the SAN's controller performs fault handling. A person of ordinary skill in the art at the time of the invention would have been motivated to perform fault diagnosis on a SAN because from line 39 of column 4 of Lewis, "For purposes of illustration only and not to limit generality, the present invention will now be explained with reference to its use in management and resolution of faults occurring in a typical computer-based local area network. However, one skilled in the art will recognize that the present invention is applicable to other types of communications networks." A person of ordinary skill in the art at the time of the invention would have been further motivated to perform fault diagnosis using a storage virtualization controller because, as disclosed in Lewis, fault diagnosis is performed centrally (Figure 1, element 18), and similarly, Reuter discloses fault handling by a central controller (e.g., paragraph 14).

Further, although Lewis in view of Reuter does not specifically disclose an alternate failure analysis module configured as a backup to the primary failure analysis module to facilitate high-availability and redundancy, failing over a process is well known in the art. An example of this is shown by Harper, see for example figure 4. A person of ordinary skill in the art at the time of the invention would have been motivated to fail over a failed/failing process because, from the abstract of Harper, "for increased software dependability… avoiding the outage."

3. Referring to claim 2, Lewis discloses loading the error pattern and associated error action into a failure analysis module (Figure 4, elements 100, 102).

4. Referring to claim 3, 20, Lewis discloses initializing a failure analysis module with the one or more predetermined error actions, one or more predetermined system error events, and one or more predetermined input-output error events associated with the storage area network (From line 3 of column 8, “Rules for rules database 114 may be determined by having domain experts explicitly specify a set of rules that match specific faults to trouble ticket data fields. Each rule is a determinator. Using knowledge engineering techniques such as the "consult/implement/test" technique previously described, these rules can be refined manually, automatically, or by a combination of automatic and manual modification as the system deals with network faults, and can change as the network changes.” From line 43 of column 7, “In order to select relevant trouble tickets from memory 50, the relevant data fields to be looked at are those that represent things such as bandwidth, network load, packet collision rate, and packet deferment rate.” From line 16 of column 1, “Faults, as used in this disclosure, may include a failure of hardware portions of the communications network, such as workstations or peripheral devices and failure of software portions of the network, such as software application programs and data management programs.”).

5. Referring to claim 4, 21, Lewis discloses the configuration and management is performed using a centralized failure analysis module (Figure 1, element 18.).

6. Referring to claim 6, 23, Lewis discloses each of the one or more predetermined error actions describes a set of operations to accommodate the occurrence of the one

or more system error events and input-output error events (Figure 4, elements 102, 104, 106.).

7. Referring to claim 7, 24, Lewis discloses the one or more error events are selected from a set of error events including predetermined system error events and predetermined input-output error events (From line 43 of column 7, “In order to select relevant trouble tickets from memory 50, the relevant data fields to be looked at are those that represent things such as bandwidth, network load, packet collision rate, and packet deferment rate.” From line 16 of column 1, “Faults, as used in this disclosure, may include a failure of hardware portions of the communications network, such as workstations or peripheral devices and failure of software portions of the network, such as software application programs and data management programs.”).

8. Referring to claim 8, 25, Lewis discloses each of the one or more system error events occurs when an error event occurs corresponding to a module within the storage virtualization controller (From line 13 of column 5, “Fault detection module 22 monitors local area network 8 via communications link 14, configuration management module 20 and communications link 24 to detect any undesirable network conditions that indicate a fault has occurred. If a network fault is detected, fault detection module 22 may automatically gather and transmit appropriate fault information via communications link 16 to fault processing system 18.”).

9. Referring to claim 9, 26, Reuter discloses each of one or more input-output error events corresponds to a communication error between the storage virtualization controller and servers or storage elements in the storage area network (From paragraph

24, “A sustained loss of communication between the controller 120 and mapping agent 110 also causes I/O operations to stop: either by making all mapping table entries revert to an active invalid state 240 or by adding additional mechanisms to suspend I/O operations until directed by the controller 120 to resume I/O operations.”).

10. Referring to claim 10, Lewis discloses the error pattern and associated error actions are specified incrementally over time without recoding (From line 14 of column 7, “Trouble ticket 60 is then stored in trouble ticket memory 50, thus adding to the system's knowledge base that may be accessed in order to resolve future communications network faults.”).

11. Referring to claim 11, Lewis discloses the error pattern is generated automatically through a logging and analysis of past error events (From figure 4, element 102.).

12. Referring to claim 19, Lewis discloses instructions in the memory when executed load the error pattern and associated error action into a failure analysis module in the memory (Figure 4, elements 100, 102, wherein the instructions are performed by and in the fault processing system.).

13. **Claims 12-16, 27-32, 34, 36 rejected under 35 U.S.C. 103(a) as being unpatentable over US 5666481 to Lewis in view of US 20020019922 to Reuter et al. and US 6336139 to Feridun et al. and US 6446218 to D’Souza.**

14. Referring to claim 12, 27, 34, Lewis discloses discloses initializing a primary failure analysis module for processing error events and error actions (Wherein a functioning process must have been initialized at some point.);

generating one or more error events responsive to the occurrence of one or more conditions of components being monitored in the network (Figure 4, elements 100, 102, 104. From the abstract, “network”.);

receiving the one or more error events over a time interval for analysis in a failure analysis module (Figure 1 shows fault detection 22 sending to the fault processing system 18, wherein events occur and are sent in a time interval.);

comparing, according to one or more rules, an arrangement of the error events received against a set of error patterns loaded in the failure analysis module (Figure 4, element 102.);

and identifying the error pattern from the set of error patterns and the error action corresponding to the error pattern to perform in response to the comparison in the failure analysis module and the one or more rules (Figure 4, element 102, 104, 106, 108.).

Although Lewis does not specifically disclose that the network may be a storage area network, error handling in a SAN is known in the art. An example of this is shown by Reuter, see paragraphs 23-26. A person of ordinary skill in the art at the time of the invention would have been motivated to perform fault diagnosis on a SAN because from line 39 of column 4 of Lewis, “For purposes of illustration only and not to limit generality, the present invention will now be explained with reference to its use in management and resolution of faults occurring in a typical computer-based local area network. However, one skilled in the art will recognize that the present invention is applicable to other types of communications networks.”

Further, although Lewis in view of Reuter does not specifically disclose the arrangement of error events is temporal, analyzing by time is known in the art. An example of this is shown by Feridun from line 4 of column 9, “Correlation rules 67 are components of or adjuncts to a given software agent. They specify a context in which to analyze or to correlate system events. Preferably, the correlation rules 67 are configured at build time for the purpose of examining a certain set of events for some observable condition. Thus, a given correlation rule 67n identifies an abstract situation of which the events it addresses are symptoms. It thus relates disparate events to a more generic problem. Typically, each rule 67 is associated with a source of events being monitored and thus a set of such rules are "correlated" to trigger a response.” Further, from line 10 of column 8, “Thus, a distributed monitor (DM) within a given local runtime environment uses "events" to convey status change(s) in monitored object(s). Events are correlated, as will be seen, using an event correlator comprising a correlation engine 65 and a set of correlation rules 67.” Further, form line 33 of column 9, “PassThrough Rules are more complex matching rules that are triggered by a specific sequence of events. This sequence can be in either specific or random order.” A person of ordinary skill in the art at the time of the invention would have been motivated to look for a temporal arrangement because, from Feridun above, “Thus, a given correlation rule 67n identifies an abstract situation of which the events it addresses are symptoms. It thus relates disparate events to a more generic problem.” Further, a person of ordinary skill in the art at the time of the invention would have been motivated

to use such an event correlator because Lewis and Reuter disclose a complex distributed environment in which events are used for fault diagnosis.

Further, although Lewis in view of Reuter and Feridun does not specifically disclose an alternate failure analysis module configured as a backup to the primary failure analysis module to facilitate high-availability and redundancyfailing over a process is well known in the art. An example of this is shown by D'Souza, from the abstract, "If the fault tolerance level is below the predefined acceptable fault tolerance level, the method also includes searching for a first suitable computer among the first plurality of computers to load another module of the software program thereon. The first suitable computer represents a computer of the first plurality of computers that does not have a module of the software program running thereon. The first suitable computer is compatible to execute the another copy of the computer program. If the first suitable computer is available, the method further includes loading the another module of the software program on the first suitable computer, registering the first suitable computer as a computer capable of servicing transaction requests pertaining to the software program after the another module of the software program is loaded onto the first suitable computer, and routing the transaction requests pertaining to the software program to the first suitable computer after the registering." A person of ordinary skill in the art at the time of the invention would have been motivated to fail over a failed/failing process because, from the abstract of D'Souza, "maintaining a predefined acceptable fault tolerance level for a plurality of software modules implementing a software program

running on a first plurality of computers coupled together in a cluster configuration in a first cluster in a clustered computer system."

15. Referring to claim 13, 28, Lewis discloses the one or more error events are converted into error event codes by a set of monitor modules monitoring the components in the storage area network (Figures 3, 7).

16. Referring to claim 14, 29, Lewis discloses the one or more error events are selected from a set of error events including predetermined system error events and predetermined input-output error events (From line 43 of column 7, "In order to select relevant trouble tickets from memory 50, the relevant data fields to be looked at are those that represent things such as bandwidth, network load, packet collision rate, and packet deferment rate." From line 16 of column 1, "Faults, as used in this disclosure, may include a failure of hardware portions of the communications network, such as workstations or peripheral devices and failure of software portions of the network, such as software application programs and data management programs.").

17. Referring to claim 15, 30, Lewis discloses each of the one or more system error events occurs when an error event occurs corresponding to a module within the storage virtualization controller (From line 13 of column 5, "Fault detection module 22 monitors local area network 8 via communications link 14, configuration management module 20 and communications link 24 to detect any undesirable network conditions that indicate a fault has occurred. If a network fault is detected, fault detection module 22 may automatically gather and transmit appropriate fault information via communications link 16 to fault processing system 18.").

18. Referring to claim 16, 31, Reuter discloses each of one or more input-output error events corresponds to a communication error between the storage virtualization controller and servers or storage elements in the storage area network (From paragraph 24, “A sustained loss of communication between the controller 120 and mapping agent 110 also causes I/O operations to stop: either by making all mapping table entries revert to an active invalid state 240 or by adding additional mechanisms to suspend I/O operations until directed by the controller 120 to resume I/O operations.”).

19. Referring to claim 32, D’Souza further discloses that there may be plural backup modules (D’Souza, from the abstract, “If the fault tolerance level is below the predefined acceptable fault tolerance level, the method also includes searching for a first suitable computer among the first plurality of computers to load another module of the software program thereon. The first suitable computer represents a computer of the first plurality of computers that does not have a module of the software program running thereon. The first suitable computer is compatible to execute the another copy of the computer program. If the first suitable computer is available, the method further includes loading the another module of the software program on the first suitable computer, registering the first suitable computer as a computer capable of servicing transaction requests pertaining to the software program after the another module of the software program is loaded onto the first suitable computer, and routing the transaction requests pertaining to the software program to the first suitable computer after the registering.”).

20. Referring to claim 36, Lewis discloses initializing a primary failure analysis module for processing error events and error actions (Wherein a functioning process

must have been initialized.);

generating, according to one or more rules, one or more error patterns automatically through logging of error events and analysis of the error events occurring in the network (Figure 1 shows fault detection 22 sending to the fault processing system 18, wherein events occur and are sent in a time interval. Figure 4, element 102.);

suggesting that the one or more error patterns generated from the analysis receive at least one error action to be performed according to the one or more rules and in the event the one or more error patterns occur on the storage area network; and associating an error action to perform in response to each of the suggested one or more error patterns generated from the analysis (Figure 4, 104-108.).

Although Lewis does not specifically disclose that the network may be a storage area network, error handling in a SAN is known in the art. An example of this is shown by Reuter, see paragraphs 23-26. A person of ordinary skill in the art at the time of the invention would have been motivated to perform fault diagnosis on a SAN because from line 39 of column 4 of Lewis, “For purposes of illustration only and not to limit generality, the present invention will now be explained with reference to its use in management and resolution of faults occurring in a typical computer-based local area network. However, one skilled in the art will recognize that the present invention is applicable to other types of communications networks.”

Further, although Lewis in view of Reuter does not specifically disclose the arrangement of error events is temporal, analyzing by time is known in the art. An example of this is shown by Feridun from line 4 of column 9, “Correlation rules 67 are

components of or adjuncts to a given software agent. They specify a context in which to analyze or to correlate system events. Preferably, the correlation rules 67 are configured at build time for the purpose of examining a certain set of events for some observable condition. Thus, a given correlation rule 67n identifies an abstract situation of which the events it addresses are symptoms. It thus relates disparate events to a more generic problem. Typically, each rule 67 is associated with a source of events being monitored and thus a set of such rules are "correlated" to trigger a response." Further, from line 10 of column 8, "Thus, a distributed monitor (DM) within a given local runtime environment uses "events" to convey status change(s) in monitored object(s). Events are correlated, as will be seen, using an event correlator comprising a correlation engine 65 and a set of correlation rules 67." Further, form line 33 of column 9, "PassThrough Rules are more complex matching rules that are triggered by a specific sequence of events. This sequence can be in either specific or random order." A person of ordinary skill in the art at the time of the invention would have been motivated to look for a temporal arrangement because, from Feridun above, "Thus, a given correlation rule 67n identifies an abstract situation of which the events it addresses are symptoms. It thus relates disparate events to a more generic problem." Further, a person of ordinary skill in the art at the time of the invention would have been motivated to use such an event correlator because Lewis and Reuter disclose a complex distributed environment in which events are used for fault diagnosis.

Further, although Lewis in view of Reuter and Feridun does not specifically disclose an alternate failure analysis module configured as a backup to the primary

failure analysis module to facilitate high-availability and redundancy, failing over a process is well known in the art. An example of this is shown by Harper, see for example figure 4. A person of ordinary skill in the art at the time of the invention would have been motivated to fail over a failed/failing process because, from the abstract of Harper, "for increased software dependability… avoiding the outage."

21. **Claim 35 rejected under 35 U.S.C. 103(a) as being unpatentable over US 5666481 to Lewis in view of US 20020019922 to Reuter, "threshold" by IEEE, and "graphical user interface" by Microsoft Computer Dictionary (herein MSCD) and US 6629266 to Harper et al.**

22. Referring to claim 35, Lewis discloses discloses initializing a primary failure analysis module for processing error events and error actions (Wherein a functioning process must have been initialized at some point.);

identifying one or more predetermined error actions and one or more error events associated with the network; specifying an error pattern based upon a combination of one or more error events in the network, and according to one or more rules, presented through a user interface with corresponding determination values (From line 48 of column 5, "Fault processing system 18 also includes a user interface module 38 coupled to fault resolution system 32 via communications link 40 and trouble-ticketing system 34 via communications link 42. User interface module 38 allows a user to edit and control proposed fault resolutions generated by fault resolution system 32 using keyboard 44." From line 3 of column 8, "Rules for rules database 114 may be determined by having domain experts explicitly specify a set of rules that match specific

faults to trouble ticket data fields. Each rule is a determinator. Using knowledge engineering techniques such as the "consult/implement/test" technique previously described, these rules can be refined manually, automatically, or by a combination of automatic and manual modification as the system deals with network faults, and can change as the network changes." From line 6 of column 10, "Critic-based adaptation module 124 allows a user, through user interface 38 to edit a proposed displayed potential solution presented by propose step 106 or to enter his or her own solution to the outstanding network fault. Critic-based adaptation is another form of adaptation that allows the system to adapt previous resolutions to novel network faults. Critic-based adaptation includes adding, removing, reordering, or replacing steps in the proposed retrieved solution. For example, considering the first retrieved trouble ticket described in connection with FIG. 6 above, a maintenance and repair person could include the data field "network.sub.-- load" and refine the solution by providing a two-place function f(F,N) that calculates the amount of adjustment based on the values of file "transfer.sub.-- throughput" and "network load".");

and associating an error action presented through the user interface to perform according to one or more rules and in response to receiving the combination of one or more error events of the error pattern that satisfy the determination value requirements (Figure 4, elements 102, 104, 106).

Although Lewis does not specifically disclose that the network may be a storage area network and that the errors are managed by a storage virtualization controller, error handling by a storage virtualization controller in a SAN is known in the art. An

example of this is shown by Reuter, see paragraphs 23-26, wherein the SAN's controller performs fault handling. A person of ordinary skill in the art at the time of the invention would have been motivated to perform fault diagnosis on a SAN because from line 39 of column 4 of Lewis, "For purposes of illustration only and not to limit generality, the present invention will now be explained with reference to its use in management and resolution of faults occurring in a typical computer-based local area network. However, one skilled in the art will recognize that the present invention is applicable to other types of communications networks." A person of ordinary skill in the art at the time of the invention would have been further motivated to perform fault diagnosis using a storage virtualization controller because, as disclosed in Lewis, fault diagnosis is performed centrally (Figure 1, element 18), and similarly, Reuter discloses fault handling by a central controller (e.g., paragraph 14).

Although Lewis in view of Reuter does not specifically disclose the determination value may be a threshold value, using thresholds to determine actions is very well known in the art. An example of this is shown by IEEE, "A value of voltage or other measure that a signal must exceed in order to be detected or retained for further processing." A person of ordinary skill in the art at the time of the invention would have been motivated to use a threshold because, from IEEE, "to be detected or retained for further processing" and further from line 9 of column 9 to line 40 of column 10, Lewis performs value judgment for exemplary throughput and load; values for which threshold limitations would clearly be beneficial in judging.

Further, although Lewis in view of Reuter does not specifically disclose the user

interface may be a GUI, GUIs are very well known in the art. An example of this is shown by MSCD, “A type of environment that represents programs, files, and options by means of icons, menus, and dialog boxes on the screen.” A person of ordinary skill in the art at the time of the invention would have been motivated to use a GUI because, from MSCD, “The user can select and activate these options by pointing and clicking with a mouse or, often, with the keyboard.”, and further, GUIs simplify the interface experience, providing obvious benefits in accessing a particular element and intuitive operation.

Further, although Lewis in view of Reuter, “threshold”, and “GUI” does not specifically disclose an alternate failure analysis module configured as a backup to the primary failure analysis module to facilitate high-availability and redundancy, failing over a process is well known in the art. An example of this is shown by Harper, see for example figure 4. A person of ordinary skill in the art at the time of the invention would have been motivated to fail over a failed/failing process because, from the abstract of Harper, “for increased software dependability… avoiding the outage.”

(10) Response to Argument

Before addressing the arguments in the order presented by Applicant, Examiner wishes to point out Applicant’s main argument which concerns the claimed “rules” and the primary reference's reference to “rules based reasoning”. Rules based reasoning is an art specific term which has understood meaning. This can be seen as applied in the reference Lewis in columns 2 and 3. Examiner does not re-interpret rules based reasoning and does agree with the explanation provided by Lewis, and the further

contrasting concept of case based reasoning as provided by Lewis.

However, Examiner notes, and further explains below, that Applicant at no point equates “rules” to rules based reasoning. Instead, Applicant describes at length the distinction that Lewis draws between rules based reasoning (RBR) and case based reasoning (CBR) and further attempts to draw the conclusion that because Lewis declines the use of RBR, that a claim constructed with the word “rules" must therefore preclude a system that uses CBR.

As the claims have failed to describe the specificity of rules beyond their mere presence in certain steps (see for example claim 1, where “one or more rules” are interjected in specifying and associating steps), Examiner must give “rules” its broadest reasonable definition as, inter alia, “a prescribed guide for conduct or action”. Indeed, the claims merely state that some action is done “according” to rules, but does not give any further specification as to how such an accommodation is achieved.

However, even as described in the specification, there is no equating of rules and RBR, and there is a further marked similarity between "rules" as used by Applicant and CBR as applied in Lewis. For example, in page 6, paragraph 11, Applicant’s specification indicates that “Rule-driven or policy based error rules can be generated without additional code using a set of predetermined error events and error actions." In other words, an error rule is generated by combining some set of events and actions, minimally one event and one action. As described, this is similar to a CBR system as described by Lewis in that a case or scenario used in CBR also has error events and prescribed actions (see, for example, Lewis, figures 3 and 4). Further similarly, the

similarity of a current trouble ticket, in Lewis, is used to retrieve past, completed cases and scenarios (the claimed “error pattern”) for use in determining a prescribed course of action (Lewis, figure 4 and abstract).

A. a. Applicant argues (page 11) that Lewis and Reuter are non-analogous art as Lewis applies to communication networks which primarily deal with data transmission whereas Reuter applies to storage area networks which primarily deal with data storage. Examiner points out that it is highly non-coincidental that “storage area network” has the word ‘network’ in the term and that further, data stored in a SAN is done so via a communication network. Although Lewis was specifically practiced in a communication network, it obviously could have been practiced in any environment with a network, and further any such complex environment could have benefited from such an error analysis system.

Applicant argues (page 11) that Lewis and Reuter are not from the same field of endeavor. This is clearly not the case as they are both related to computers and computer networking. Applicant further argues that Lewis is not reasonably pertinent to the problem to be solved. As indicated above, Examiner believes that any complex, communicative environment could have benefited from the trouble ticketing system of Lewis. However, Lewis itself indicates that the approach is non-limiting. From line 59 of column 10, “one of ordinary skill in the art will recognize that the present invention is applicable to networks other than local area networks… can be used with communication networks fault management systems other than trouble ticket type systems, such as spread sheet systems or database systems.”

Applicant argues (page 12) that the network faults of Lewis are errors since they result from a failure of hardware or software whereas Reuter specifically is concerned with the management of mapping tables for SAN. Examiner did not rely on the specific workings of the SAN of Reuter. Examiner cited Reuter primarily to show a SAN environment comprising a storage virtualization controller, as necessitated by the claim language. Regardless, Examiner notes that any such error/fault/failure in a SAN, such as the one disclosed in Reuter, also produces symptoms, not unlike those used in Lewis, which may be used in fault diagnosis. It is Examiner's view that Applicant has failed to claim in any meaningful way the significance of SAN. As such, it is treated as an obvious variant to a well known technique.

Applicant argues (page 14) that there must be an apparent reason to combine the known elements in the fashion claimed. Examiner presents such apparent reasoning in the teaching, suggestion, motivation of the rejections. This is otherwise known as the Graham v. Deere factors, which Examiner did show in constructing every rejection based on combination.

Applicant argues (page 14) that Examiner has not provided sufficient reasoning to combine, only stating that Harper should be combined with Lewis and/or Reuter for "increased software dependability…avoiding the outage" and that these are "conclusory statements" that merely repeat overarching goals in Harper and provide no "explicit connection" to either Lewis and/or Reuter. Examiner is, in turn, unclear just what would have satisfied Applicant with regards to reason for combination. Regardless, Examiner has met the burden of a prima facie case by providing references that meet the

limitations and supplying Graham v. Deere factors for combination. Had Examiner found references that provided the level of “explicit connection” required by Applicant, Examiner suspects that a 103 rejection would not have been necessary.

Applicant argues (page 14) that Lewis is not actually about making a network more dependable but concerns a trouble-ticket management system. Examiner points out that it is for both. Notably, Lewis discloses a trouble ticket management system *for* making a network more dependable.

b. Applicant argues (page 16) that Lewis does not teach “initializing a primary failure analysis module for processing error events and error actions”. Examiner pointed to the fact that Lewis has a functioning, operating failure analysis module that processes error events and actions, and therefore, logically, inherently, it must have been initialized at some point. Applicant addresses this as a "conclusory statement" based on common sense. This is not common sense, but something even baser as it relies on a simple understanding of the technology and what must be there, i.e., inherency. Even outside of the technology, human beings are hard put to indicate how something could exist without having had an initial state. In the area of computers, it is no mystery that an operating process must have been initialized at some point.

Applicant argues (page 17) that Lewis does not identify error events but instead identifies cases or scenarios. Examiner points to the precursory statements given above regarding RBR versus CBR versus rules. Here, Applicant relies heavily on a single word “rules” to imply that the CBR of Lewis may not be used. However, as claimed, Lewis still applies. Specifically regarding identifying **one** or more predetermined error actions and

one or more error events, Examiner cited figure 4, elements 100, 102, and 104, which indicates that fault information, and past (predetermined) trouble tickets (which include events and actions) are identified. There is nothing in the claims that differentiate between events and actions as specified in a case or scenario (Lewis) and simply identifying events and actions. Examiner further emphasizes that in the "identifying…" limitation, Applicant does nothing more than to say that these events and actions are identified. Applicant makes no effort to claim by what or how. Furthermore, the "predetermined error actions and… error events" of this limitation are not referred to as an antecedent further in the claim. Subsequent events and actions are referred to separately.

Applicant argues (page 17-18) that figure 4 of Lewis teaches "specifying…" and "associating…" but provides no detailed reasoning. Applicant apparently ignores the more specific figure elements that Examiner referred to in addressing each of those limitations. As shown in the rejection, Lewis specifies through figure 4, elements 100 and 102, where Lewis shows that past/completed trouble tickets (error patterns) are chosen based on fault information (error events). Here Examiner further notes that at least from Lewis's abstract, Lewis further indicates that (with emphasis) "Completed trouble tickets are stored in a library and when an outstanding trouble ticket is received, the system uses at least one **determinator** to correlate the outstanding communications network fault to data fields in the set of data fields of the trouble ticket data structure to determine which completed trouble tickets in the library are relevant to the outstanding communications network fault. The system retrieves a set of completed trouble tickets

from the library that are similar to the outstanding trouble ticket and uses at least a portion of the resolution from at least one completed trouble ticket to provide a resolution of the outstanding trouble ticket." Such a determinator, that matches based on similarity, bears marked similarity to whatever it is that Applicant believes a rule should be. Even if it does not, it clearly meets the common meaning of the term, "a prescribed guide for conduct or action".

Further, regarding "associating", Examiner referred to figure 4, elements 102, 104, and 106. Here Lewis shows that retrieved, completed trouble tickets are retrieved for their resolutions so that they may be adapted to an outstanding fault of the network. Here again, Applicant relies on the scantily claimed "rules" in attempting to differentiate. However, Lewis shows action "according to rule" by adapting the retrieved resolution and proposing potential resolutions. Clearly such proposed resolutions are thusly "associated" via "rules".

In sum, Lewis identifies error events and actions by matching a current trouble ticket, which itself has error events, to past trouble tickets, which have error events and error actions. Those error actions are in turn used or adapted for use. See figures 3 (particularly fields 62L-62P) and 4.

Applicant argues (page 18) that Harper does not have an application on both machines but only allows one application to run at a time just in case it is buggy and consumes too many resources. Examiner points out that Applicant merely claims that an alternate failure analysis module configured as a backup to the primary is initialized. Clearly there is nothing that requires that these be running at the same time.

c. Applicant argues (page 20) that Lewis teaches away from RBR as disclosed in the background of Lewis, in favor of CBR. Examiner points to the precursory response presented above. Further, Examiner agrees that Lewis uses CBR in favor of RBR. However, Applicant has not claimed RBR, but merely “according to rules” and no more. Furthermore, as shown above, Lewis does have "rules", or at least elements such as the determinator and resolution adapting process that function as rules. Even if these are not “rules” that Applicant would like, they are still rules by the definition of what a rule is, “a prescribed guide for conduct or action”. As Applicant has made no effort to claim beyond “according to rules”, Examiner must give "rules" its broadest, reasonable definition. Further still, even though Lewis uses CBR, it still bears a marked similarity to Applicant’s method of failure analysis in that a current event(s) (fault information) is matched with other events (completed, past trouble tickets), so that matching events may be used to determine a course of action (adapt, propose resolutions).

Applicant argues (page 20) that Lewis “states quite strongly that using general rules as recited in claim 1 should be avoided". Lewis clearly states no such thing as Lewis did not have the ability to predict that Applicant would be claiming claim 1. What Lewis does state is that Lewis would not be using RBR. Even though RBR clearly has RBR rules, there is no indication that those RBR rules are the same as the broadly claimed “rules” of claim 1.

Even had Applicant claimed RBR, which the specification does not explicitly provide for, Lewis makes clear that the use of RBR is not novel and was known as Lewis discloses it in Lewis's background section, a patent filed in February of 1993.

d. Applicant provides no new arguments.

B. a. Applicant argues (page 22) that Examiner has not provided reasoning or support, instead referring to Examiner's adherence to Graham v. Deere as "conclusory statements". Examiner points to every combination made in the rejections of record in support of the position that even if these were to be termed “conclusory statements”, they are still clearly prima facie cases for obviousness in accordance with proper examining procedure, complete with reasoning and rationale.

Applicant argues (page 23) that “Examiner merely repeats an overarching goal found in the abstract of D’Souza that states ‘maintaining a predefined acceptable fault tolerance level for a plurality of software modules…’” and labels this, again, as a “conclusory statement”. Again, Examiner has difficulty in understanding just what it is that Applicant is demanding. Merely labeling the Graham v. Deere factors as “conclusory statements" does not negate the fact that they are still Graham v. Deere factors. Labeling a motivation as an “overarching goal” does not make it any less of a motivating factor. As above, Examiner has shown a prima facie case of obviousness in accordance with proper examining procedure. Each of the combinations made in rejecting the claims are complete with teaching, suggestion, and motivation.

b. Applicant again argues (page 24) as Applicant did on page 16, that Lewis does not teach “initializing a primary failure analysis module for processing error events and error actions”. Again, Examiner points to the fact that Lewis has a functioning failure analysis module that processes error events and actions, therefore, logically, inherently, it must have been initialized at some point. Applicant addresses this a "conclusory

statement" based on common sense. This is not common sense, but something even baser as it relies on a simple understanding of the technology and what must be there, i.e., inherency. Even outside of the technology, human beings are hard put to indicate how something could exist without having had an initial state. In the area of computers, it is no mystery that an operating process must have been initialized at some point.

Applicant argues (page 25) that since Lewis teaches CBR, it cannot be combined with the temporal correlation rules of Feridun. First, again, Applicant incorrectly assumes that simply because a word “rule” is used in a reference, in this case Feridun, it must be the same type of rule, or read in the same context, as the “rule" used in the claim. As they are not based on the same specification, this is an unreasonable assumption. Secondly, it is not clear how Applicant arrives at the conclusion that simply because Lewis uses cases or scenarios that those cases or scenarios cannot specify temporal occurrences. To analyze something on a temporal basis is very well known in the art, of which Feridun is an example. Lewis shows (for example, figure 3) that there are plural events or states that are arrived at which go towards defining a trouble ticket. Feridun shows that correlation based on temporal order aids in anaylsis.

Applicant argues (page 25) that reason to combine must be based on design need, market pressure, and having a finite number of identified, predictable solutions in technical grasp. Applicant cites KSR as supposedly corroborating this stance. However, it should be clear that KSR teaches flexibility and provides certain examples of how flexibility can be applied. KSR does not provide a rigid set of rules that must be adhered to in determining reason to combine. In fact, KSR goes so far as to say that Graham v.

Deere may be too rigid. Regardless, as Examiner has applied the more rigid Graham v. Deere factors in making the rejections, it should be clear that Applicant's argument is moot.

Applicant further argues (page 25) that Feridun provides a correlation engine for processing correlator rules that may be programmed without any clearly finite limitations. Here it should be apparent that Applicant has failed to grasp either the "finite number of predictable solutions" referenced in KSR, or the combination of Feridun with Lewis. Here, there are two (a finite number) combinations between Feridun and Lewis, those two being either correlation with temporal analysis or correlation without temporal analysis. It is not clear where Applicant is drawing this supposedly infinite number of combinations.

c. Applicant provides no new arguments.

d. Here again, Applicant argues (page 26-28) that Lewis teaches away from the use of RBR. Again, Applicant does not teach the use of RBR either, but scantily claimed "rules".

C. Here again, Applicant argues initializing (this was already addressed twice previously), CBR versus RBR versus rules (this was addressed repeatedly and throughout), and Reuter not concerning CBR but page faults (this was addressed previously concerning non-analogous art).

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/Gabriel L. Chu/

Primary Examiner, Art Unit 2114

Conferees:

/STB/

SPE, Art Unit 2114

/RWB/

SPE, Art Unit 2113